

# Dokumentacja projektu „Gra w Wisielca



## Spis treści

Dokumentacja projektu „Gra w Wisielca .....	1
Opis ogólny.....	4
Struktura projektu .....	4
Opis poszczególnych plików.....	5
Główne funkcjonalności .....	6
Logika gry .....	6
Przykład użycia.....	7
Technologie .....	7
Licencja.....	7

## Opis ogólny

Projekt stanowi aplikację typu "Gra w Wisielca", która pozwala na grę w klasyczną grę słowną, w której gracz stara się odgadnąć ukryte słowo, zgadując litery. Gra wykorzystuje bibliotekę **Flet**, która umożliwia tworzenie aplikacji webowych desktopowych i mobilnych w Pythonie.

Aplikacja zawiera różne widoki: ekran główny, ustawienia, informacje o grze, a także widok samej gry. Gra oferuje możliwość wczytania własnych słów, ustawienia maksymalnej liczby błędów, a także umożliwia rozpoczęcie rozgrywki.

## Struktura projektu

Projekt składa się z kilku głównych komponentów:

1. **Widok główny (index\_view.py)** – Strona, która umożliwia rozpoczęcie gry lub przejście do ustawień.
2. **Widok informacji (info\_view.py)** – Ekran z opisem gry i zasadami.
3. **Widok ustawień (settings\_view.py)** – Ekran umożliwiający wybór pliku ze słowami oraz ustawienie maksymalnej liczby błędów.
4. **Widok gry (play\_view.py)** – Ekran, na którym odbywa się sama rozgrywka.
5. **Backend** – Moduły odpowiedzialne za logikę gry, zarządzanie stanem gry i przekazywanie informacji między widokami.

## Opis poszczególnych plików

### **index\_view.py**

Widok główny aplikacji, który pozwala użytkownikowi przejść do gry lub ustawień. Zawiera dwa główne przyciski:

- **Graj** – przechodzi do widoku gry.
- **Ustawienia** – przechodzi do widoku ustawień.

### **info\_view.py**

Widok, który zawiera informacje o grze. Zawiera zasady gry, jak grać, cel gry oraz instrukcje do ustawienia własnych słów i liczby błędów. Jest to pomocny ekran dla gracza, który nie jest jeszcze zaznajomiony z zasadami gry w wisielca.

### **settings\_view.py**

Widok, który pozwala użytkownikowi na:

- Wczytanie własnego pliku ze słowami, które będą używane w grze.
- Określenie maksymalnej liczby pomyłek, które gracz może popełnić podczas gry.
- Wybór pliku odbywa się za pomocą kontrolki FilePicker.
- Maksymalna liczba pomyłek jest przechowywana w sesji i używana podczas gry.

### **play\_view.py**

Widok gry, w którym gracz wchodzi w interakcję z aplikacją. Zawiera:

- Wyświetlanie słowa, które gracz musi odgadnąć, w postaci podkreśleń.
- Wprowadzanie liter przez gracza.
- Pokazywanie liczby pomyłek, które zostały popełnione.
- Informację o dostępnych słowach, które można zgadywać.
- Po zakończeniu gry (wygranej lub przegranej) umożliwia rozpoczęcie nowej gry.

### **Backend (CurrentRoute.py, Game.py, Snack.py, Settings.py)**

- **CurrentRoute.py** – Zarządza obecnym stanem trasy (czyli informacją, na jakiej stronie użytkownik się znajduje).
- **Game.py** – Zawiera logikę gry, w tym funkcje do losowania słów, sprawdzania liter oraz obliczania liczby błędów.
- **Snack.py** – Definiuje funkcję do wyświetlania snack barów, czyli krótkich powiadomień o zdarzeniach, takich jak błędy lub sukcesy.

- **Settings.py** – Obsługuje ustawienia, w tym wybór pliku z słowami i zapisywanie ustawienia maksymalnej liczby pomyłek.

## Główne funkcjonalności

**Rozgrywka:** Gracz odgaduje litery w ukrytym słowie. Za każdą literę, która nie pasuje do słowa, liczba błędów rośnie. Gra kończy się, gdy gracz zgadnie całe słowo lub przekroczy dozwoloną liczbę błędów.

**Ustawienia:** Gracz może:

- Wczytać plik z własnymi słowami do gry.
- Ustawić maksymalną liczbę pomyłek.

**Wybór pliku:** Gracz może wybrać plik tekstowy zawierający słowa, które będą używane w grze. Słowa są oddzielone przecinkami.

**Zasady gry:** Gracz może zapoznać się z zasadami gry, jak również instrukcjami dotyczącymi interakcji z aplikacją.

## Logika gry

**Losowanie słowa:** Po rozpoczęciu gry, losowane jest słowo z dostępnej puli słów (załadowanej z pliku). Słowo jest reprezentowane przez podkreślenia, a litery, które gracz zgaduje, są odsłaniane w odpowiednich miejscach.

**Sprawdzanie liter:** Gracz wpisuje literę, która jest sprawdzana w kontekście poprawności (czy to litera, czy była już użyta). Jeśli litera jest w słowie, zostaje odkryta, w przeciwnym przypadku liczba błędów rośnie.

**Warunki wygranej i przegranej:** Gra kończy się, gdy:

- Gracz zgadnie wszystkie litery w słowie (wygrana).
- Gracz popełni maksymalną liczbę błędów (porażka).

**Przyciski i interakcje:**

- Przycisk "Graj" rozpoczyna grę.
- Przycisk "Ustawienia" umożliwia przejście do ustawień gry.
- W trakcie gry wyświetlane są informacje o liczbie pomyłek i dostępnych słowach.

## Przykład użycia

1. Uruchom aplikację.
2. Na ekranie głównym kliknij "Graj", aby rozpocząć grę lub "Ustawienia", aby dodać własne słowa.
3. W ustawieniach wybierz plik ze słowami lub pozostaw domyślne słowa. Ustaw liczbę dopuszczalnych błędów.
4. Wróć do gry i zgaduj litery, aż zgadniesz słowo lub przekroczysz dozwoloną liczbę pomyłek.
5. Po zakończeniu gry, kliknij "Zaczynij od nowa", aby ponownie rozpocząć rozgrywkę.

## Technologie

- **Flet:** Biblioteka do tworzenia aplikacji webowych w Pythonie.
- **Python 3.x:** Język programowania użyty do stworzenia aplikacji.

## Licencja

Projekt jest dostępny na licencji MIT. Można go używać, modyfikować i rozpowszechniać zgodnie z warunkami licencji MIT.